Compucolor® Corporation

Vol.II.6 AUG'79 C O L O R C U E $1.00

EDITOR'S LETTER

As was announced in June's issue of COLORCUE, editor Susan Sheridan left Compucolor and moved to the west coast. Her presence here will be sorely missed but we look forward to hearing from her occasionally. As every cloud has a silver lining, I am delighted to have been given the opportunity to become your new editor. I look forward to meeting and hearing from many of you in the upcoming months. Changes in editorship are sometimes awkward and right now I need your advice and support in large quantities.

COLORCUE was started as a way to provide additional technical support from Compucolor Corporation as well as to provide an open forum for our growing CCII user group. Naturally, I want this publication to continue to reflect your influence.

As I said, I need your help. One of the first things you could do would be to complete the short survey enclosed in this issue. I would like to reformat COLORCUE somewhat, but before I do I need to know what kind of publication would be most useful to you. Your comments will show me where COLORCUE has been and in what direction we should be going. The survey has been postage prepaid and can be folded in half, stapled or taped shut, and dropped in the mail as is. Please take some time and do this for me today.

I regret that, due to the transfer of the publication, the July issue will not be forthcoming. To make up for the missing issue, all current subscriptions will be extended by one month.

Feel free (in fact, obligated!) to submit articles and share ideas or problems with us from time to time. As an additional incentive, we are willing to trade a free subscription to COLORCUE or two Sof-Disk albums for any articles we publish. If I may be of any personal assistance, please get in touch with me

c/o Colorcue
100 Northcreek, Suite 250
3715 Northside Parkway, N.W.
Atlanta, Georgia 30327
(404) 261-3003

I look forward to writing for you.

Cathy Abramson
Editor

****

This Issue's MENU

****

"ALIEN" SPECIAL EFFECTS PRODUCED ON AN INTECOLOR COMPUTER

If you've experienced "Alien", you've seen stills made from graphics generated on an Intecolor computer. Intecolor is manufactured by Compucolor's parent company, Intelligent Systems Corp. The stills for "Alien" were produced by a British special effects lab, System Simulation.

You can shoot your own special effects stills for business or pleasure on your Compucolor II. The following article by Dr. A. W. Grogono of Syracuse, New York, describes how he photographs slides from his Compucolor.

PHOTOGRAPHING THE COMPUCOLOR SCREEN

By A. W. Grogono, M.D.
Department of Anesthesiology
Upstate Medical Center
Syracuse, New York 13210

For many months I have made all my lecture slides by photographing images prepared on a Compucolor or an Intecolor Computer. These notes briefly outline how to make the image, how to save it, and how to photograph it.

Preparing an Image for a Lecture Slide: The best slides are simple. Text should be limited to less than seven lines with few words per line. Such slides are most easily typed in "CRT mode"; this allows full cursor control. Graphs and diagrams should be uncomplicated and may be prepared in BASIC.

Saving Images: For convenient photography, save the images on the disk. This can be done by typing the instruction using a color choice which will be invisible; for convenience write a program called "MENU" which will "AUTO-LOAD" and perform this function for you; most sophisticated of all, use the elegant "SCREENSAVE" program available from Compucolor. This program prepares disks and saves, displays or erases images as required.

Camera: A single lens reflex with accessory lenses or extension tubes is almost essential or parallax error, focusing and filling the whole slide will all be problems. Kodak recommends a red color correcting filter (Kodak CC40R) for photographing television screens. I have tried the red filter with many films and test exposures and have found that it makes no appreciable difference.

Film and Exposure: Slow speed fine grain films are best. I use Ektachrome ER (ASA 64) for local processing, and Kodachrome KM (ASA 25) for processing by Kodak. Satisfactory exposures are as follows:

Film	ASA	f Stop	Exposure
Ektachrome ER	64	4.0	0.5 seconds
Kodachrome KM	25	4.0	1.0 seconds

Colors: Very few color combinations can be relied upon to project effectively. Do not use dark characters on light grounds, for instance, red on yellow. Do use the reverse, yellow on red. Avoid combinations which employ changing two or three color guns. Pale blue on red may appeal to you, but if the "hole" left by the red gun misses the characters projected by the green and blue guns the result is a mess. Most colors may be used on black but red and dark blue will be slightly too dark at the suggested exposures. The combinations listed below employ only a change of one gun and have been carefully tested:

Yellow on Red
Cyan on Dark Blue
White on Magenta
Green on Black

Photography: Employ a tripod, a cable release, and use total darkness. Reflections from lamps, tripod, and windows will spoil your slides. Make two copies each: one to use and one to lose.

Film Processing Warning: If you have slides with text on black background, the automatic film cutter will fail to recognize the frame line and will cut the frame in half. As has twice happened to me, Kodak may return your set of slides containing half the image on one slide and half on the next: in my case, this was amusing for everyone but the lecturer. Both Kodak and the audience were extremely kind but the lecture was markedly different from the one intended.

Note on Screensave Program: Children of about eight years and up readily learn to enjoy making pictures on Compucolor Screens in CRT Mode. Their pleasure is enhanced if they can save their art work merely by pressing "AUTO".

BASIC UTILITIES SOF-DISK

The BASIC Utilities Sof-Disk is an exciting new addition to our many Compucolor software offerings. This diskette contains a variety of utility programs to assist you in writing and editing BASIC programs. The following programs and features are available on the BASIC Utilities disk:

FRED 16, 32 - FRED stands for "Friendly Editor" and is the most useful program on the disk. It comes in two versions, one for 16K machines and the other for 32K machines. Once FRED is loaded (into high memory) you can switch back and forth between the Editor and BASIC by using an escape sequence. When operating in the Editor, six editing commands are available:

L - lists BASIC starting at a selected line number.

L AND SEARCH - is a variation of the List command which allows you to search for the occurrence of any variable, string, or command and list the lines containing them. This capability is excellent for debugging your programs.

E - edit any line number.

C - copies a line to a new location. The old line remains and the new line can be edited.

D - deletes a range of line numbers.

I - inserts a line number.

A - automatically numbers inserted lines. You input the starting line number and the selected increments.

B - transfers control back to BASIC. The edited program can now be saved or run. If other changes are required, an ESC sequence transfers back to the editor.

RENUM - renumbers a BASIC program. To renumber, enter the starting line number and the amount of the desired increment. A section of the program can also be renumbered by entering the starting and ending line numbers in the old program.

MERGE - merges two BASIC programs together. Lines and numbers are automatically interlaced to create a new program.

COMPAC - removes all unnecessary blanks from a BASIC program.

REMPAC - removes all remarks from a BASIC program.

As you can see, this Sof-Disk contains some very useful programs! Anyone interested in programming will find this diskette an absolute necessity. Basic Utilities costs a low $29.95. For further information please visit your local Compucolor dealer.

USER SOFTWARE FILE

We receive numerous calls from Compucolor users who want to know if we'd be interested in software they have written. The answer is yes, of course. We are always looking for quality programs. Our major interest at this time is in applications and educational programs. Most programs which have been submitted to us thus far have been games. Although we are still interested in games, we must be highly selective before we accept anymore.

Most individuals who submit programs are interested in making a profit. If your program is accepted, you may:

1. trade it for Compucolor hardware or software (this is the most common agreement made);
2. accept a cash purchase for your program, the amount paid depends on the quality and potential demand for the program;
3. accept a royalty payment. This would be done only in very special cases. All programs on the Sof-Disk album must be written by the same person in order to be eligible for a royalty agreement.

Programs are evaluated according to the following criteria:

1. potential demand for the program,
2. quality of the program,
    a. use of graphics
    b. ease of use
3. uniqueness of program -- we frequently receive several programs which do the same thing, and finally
4. documentation. If we can't quickly determine how to use the program, it will be returned.

All programs submitted for consideration should include the following:

1. Sof-Disk containing the program (your name, address, and phone number must be written on the disk label)

2. all documentation

3. a cover letter with a brief explaination of the program and what it does. The cover letter should include the following statement:

   "I, (your name) certify that the above program was written and created by me and that rights to this program have not been transferred to another party."

We encourage you to share your programming efforts with us and other Compucolor users. Send your special applications to:

Compucolor Corporation
P. O. Box 569
Norcross, Georgia 30071
Attn: User Software File

ARRAYS

Many beginning programmers have difficulty understanding the concept of arrays. An array is a powerful tool which can greatly simplify a program. In fact, many programs would be just about impossible to write without the use of arrays.

Here's an example. Simply stated, an array is a table of information. Let's suppose that we wanted to keep track of how many of each of 30 items of inventory are in stock. It would take 30 variables to keep track of the quantity of information without an array. By using an array, we can refer to all 30 variables by referring to only one variable. Variables of this type are called <u>dimensioned variables</u>.

A dimensioned variable is one which can have more than one value assigned to it. To distinguish one value of the variable from the next, we use what are called <u>subscripts</u>. The subscripts are always enclosed in parentheses following the variable and are assigned in sequential order. For example:

A(1)	refers to the first variable in the array
A(2)	to the second
A(3)	to the third, and so on.

In Compucolor BASIC the variable "A" and the dimensioned variable "A" are two different variables. They are differentiated by the existence, or lack of existence, of a subscript following the variable.

A	- normal variable
A(J)	- dimensioned variable.

To show how useful an array can be, let's attempt to find the total number of items in our inventory. We'll make the assmption that the quantity of each variable already exists in memory.

<u>Without Arrays</u>

Let's use A0 to A9, B0 to B9, and C0 to C9 to represent the 30 variables.

```
100  A=A0+A1+A2+A3+A4+A5+A6+A7+A8+A9
110  B=B0+B1+B2+B3+B4+B5+B6+B7+B8+B9
120  C=C0+C1+C2+C3+C4+C5+C6+C7+C8+C9
130  T=A+B+C
140  PRINT "TOTAL ITEMS IN INVENTORY=";T
```

<u>With Arrays</u>

```
100  T=0
110  FOR I=1 TO 30
120  T=T+A(I)
130  NEXT I
140 PRINT "TOTAL ITEMS IN INVENTORY=";T
```

In the above example, the length of the two programs are about the same. However, suppose there were 1000 items instead of 30. Without an array, our program would be over 30 times larger (assuming you could come up with 1000 unique variables). By using an array, the <u>only</u> change in the program would be to replace line 110 with: FOR I=1 TO 1000.

Subscripts in an array are always enclosed in parentheses and may also be expressed as a variable. The subscript can also be the result of a calculation, for example:

```
     R=D(I+2)*7
     T=D(J+I/2) + R(K-1)
     D(3)=A(D(I)) +A(D(I+1)+3)
```

Before a variable can be used as an array, it must be dimensioned. This is done using the <u>dimension statement</u> (DIM). To create an array for our 30 inventory items, we would use:

```
     110  DIM A(30)
```

If a subscripted variable is not dimensioned prior to its use in a program, it will automatically be dimensioned to 10. If we had not dimensioned "A" in our example, we would get an error when we started to use the 11th element of the array. The error message would read:

```
     BS ERROR IN 120     (bad subscript)
```

A variable is actually dimensioned one element longer than is indicated in the dimension statement. This is because the first element of the array always has the subscript "0" (A(0)). Many programmers ignore the zero element to simplify programming logic or use it to store something else. A frequent use of the zero element is to store the number of entries in the array. In the following example, we use it to store the total number of items in the inventory.

```
100  A(0)=0
110  FOR I=1 TO 30
120  A(0)=A(0) +A(1)
130  NEXT I
140  PRINT "TOTAL ITEMS IN INVENTORY =";A(0)
```

Thus far, we have only considered single dimensioned arrays. It is possible to dimension any number of dimensions. For example: DIM A(30), D(5,12),C(3,2,7),F(7,2,3,4). The size of the array is limited only by the amount of available memory. The number of elements in an array is the product of the dimensions plus one (the zero element). The amount of RAM required is four times the number of elements (each variable takes four bytes):

Array	Total Elements	RAM Required
A(30)	30+1=31	124
D(5,12)	(5+1)*(12+1)=78	312
C(3,2,7)	(3+1)*(2+1)*(7+1)=96	384
F(7,2,3,4)	(7+1)*(2+1)*(3+1)*(4+1)=480	1920

A multidimensioned array might be used to keep the following table:

Inventory Item No.	Number In Stock	Unit Cost	Reorder Quantity
1	5	.20	2
2	3	1.54	5
3	7	.47	6
4	8	3.58	12

and so forth to total our 30 items. We would dimension our array as A(30,3) -- actually A(29,2) would work just as well and save some memory. The number of items in stock for item 1 would be A(1,1), the unit cost of item 4 would be expressed as A(4,2), the reorder quantity of item 2 would be A(2,3).

An array can also be saved to disk or loaded from disk. This is accomplished by using the variable name followed by ".ARY".

```
SAVE "A.ARY"   or.
LOAD "A.ARY"
```

The only way to fully comprehend programming with arrays is to sit down at your Compucolor and try it. You'll discover that the array can be a powerful and useful tool for almost any program. Let's hear it for arrays -- Hip, Hip, Array!

RANDOM FILES

This is the first of two articles on the Random File System. We will begin on a basic level, and describe advanced random file applications in the second. Since the Random File System is one of the Compucolor's best features, we want all our users to be able to take full advantage of it.

What do all these "file" statements mean? File statements let the Compucolor take a disk with "blank" storage space and organize it to refer to a given set of records. In other words, the name for the defined set is the name you give to the file. A record is information within the file. The example found in the array article is a good starting point. In explaining records, let's set up a record in the M array with the number of days in each month as well as several other items. We will have the phone bill for the month N in PH(N), the power bill in PO(N) and the credit card bill in CC(N). So we have

```
M(N)            PH(N)           PO(N)           CC(N)   - 12 times.
```

We could store them any old way, but this is not very efficient. I am sure all of us have searched through a drawer full of papers and wished we had put them in folders. A "record" is the computer's equivalent to a folder, so we will have a folder, or a record, for each month. If you placed the bills in a folder, you would probably just stick in the phone, power bill and credit card bills in any order because when you pulled them out of the drawer you could tell the difference just by looking at them. The computer is dumb unless you teach it how to be smart. It is also very methodical. Because the computer does the same thing in the same way each time you run the program, if the first item in the folder is the phone bill, it will always be the first processed. The computer does not need the extra information you use to decide which bill is which, the computer just needs to know which one comes first. So, to the computer

M(N), PH(N), CC(N), PO(N)   is valid

as is PH(N), CC(N), M(N), PO(N)

However, if you put January's records into the record in the first format, and February's in the second, you would have to have some extra information (the equivalent of knowing what the bill looks like when you pull it out of a paper folder) so that the computer will know which is which. The data may be entered in whatever order you choose as long as it is entered in the same way each time. "GET" them in the same order that you "PUT" them in and the Compucolor will not confuse them. For these type applications, I always use the same statement and just exchange the term "GET" for "PUT".

(FRED the editor makes this easy!)

Now, let's set up the file. Each number (and that includes any number on the CCII) takes up four bytes. Let's also make room for a note, or record, in this folder. To make room for this note, we will allow 16 bytes or characters for it and we will have the note in the string A$(N). Add it all up (4 bytes per number * 4 numbers + 16 bytes for the note) and we get 32 bytes for each month's folder. If each record is a folder then all the folders together become a file.

You can set up a filing system on the Compucolor by typing in one line! You must tell the Compucolor four things:

1) the name of the file (this is reasonable)
2) the number of folders (ditto)
3) the size of each folder (still ok)
4) the blocking factor (what??)

What is a "blocking factor"? As we discussed above, it is best to store the minimal amount of information on the Compucolor. Disk storage is made up of 128 byte sections. For convenience and clarity, we will call them pages but the jargon is "sectors". In order not to waste space, we want to fill these pages as full as possible. The "blocking factor" is the number of records we want to fill a page. Using our example with 32 bytes to a record, four would fit in a 128 byte page and waste no space. There is something else to consider too. Like paper and folder filing, the Compucolor must get at least one whole page from the disk at a time. Getting this page from the disk can be thought of as walking downstairs to the filing cabinet.

The blocking factor determines how many folders (records) to treat as one group. It also specifies how many records the Compucolor gets on each trip to the disk. The programming manual tells you that changing the blocking factor may have great effect on the speed of your program. In our example, if we used a blocking factor of 8, each 128 byte page would contain only 32 bytes -- 96 bytes would be wasted.

Imagine some poor office worker with 100 files to go over and having to walk downstairs to get each one! The blocking factor tells the Compucolor to bring back a fixed number of records on each trip to the disk. The office worker may not have room on his desk for all 100 (or 1,000 or 10,000) folders, but he would sure bring as many folders as he could carry on each trip! Memory is the Compucolor's desk space. Use it efficiently. Set up the blocking factor so that the Compucolor has as much desk space as possible. As the Compucolor begins to search for information it will first see if the record is in memory (folder on the desk) or if it needs to go to the disk to get it. If the Compucolor is running out of room on the desk (memory) and needs an additional folder, it will close the folder that has not been looked at for the longest time and carry it back to the filing cabinet (disk). (The File "C" command says clean off your desk and put all the folders up.) All this makes the computer sound like a dependable, hardworking, and efficient clerk who is so dumb that he will spend all his time walking up and down the stairs if you let him.

Now, back to establishing our filing system. The file we are trying to set up has records of 32 bytes each. Four records fill a sector (page). We need 12 for one year. (12*32=384 bytes, that is not too much, so let's put four years on this file ...4*384=1536 bytes). OK, keeping four years on the file means a total of 48 records. Let's give 12 records to a block (12*32=384 bytes or three pages). This does not take up much memory and we will be able to keep information about a whole year's bills before the Compucolor has to go back for more! Let's see, do we know all we need to tell the CCII to set up a file?

Name? Let's call it "BILLS". (6 characters or less, that's ok)
Number of Records? 48
Record Length? 32
Blocking Factor? 12

Now, how do we actually do it? We could do it as part of the program however, this would create a new file each time we ran the program if we were not careful. However, we can do it from the keyboard! All we need to do is type the file "N" statement in Command Mode.

Here it is:

```
FILE "N","BILLS",48,32,12
```

Some other possibilities are:

FILE "N","BILLS",48,32,1	--	This wastes considerable disk storage space and would mean 12 trips to the disk to look at each month.
FILE "N","BILLS",48,32,48	--	This would mean only one trip to the disk, but would require lots of RAM since the whole file world would load at the same time.
FILE "N","BILLS",48,32,4	--	The minimum blocking factor to get the maximum use of available disk space. Requires only 128 bytes of RAM for file buffer.
FILE "N","BILLS",48,32,5	--	Not a good blocking factor since it takes 2 sectors (pages) to hold 5 records and wastes 96 bytes for every 2 sectors used.

Watch for Part 2 - "Advanced Applications -- Random Files" in next month's issue of COLORCUE.